

1 **In the Claims**

2 Claims 1, 5, 16, 18, and 20-29 are amended.

3 Claims 41 and 42 are new.

4 Claims 1, 3-16, 18-42 are pending and are listed below:

5
6 1. (Currently Amended) A software architecture for a distributed
7 computing system comprising:

8 an application configured to handle requests submitted by remote devices
9 over a network; and

10 an application program interface to present functions used by the
11 application to access network and computing resources of the distributed
12 computing system, wherein calls to the application program interface are handed
13 to a common language runtime layer that can translate Web applications written
14 in different languages into an intermediate supported language, the application
15 program interface comprising various types related to constructing user
16 interfaces, ~~wherein the various types comprise:~~

17 ~~classes which represent managed heap allocated data that has~~
18 ~~reference assignment semantics;~~

19 ~~interfaces that define a contract that other types can implement;~~

20 ~~delegates that are object oriented function pointers;~~

21 ~~structures that represent static allocated data that has value~~
22 ~~assignment semantics; and~~

23 ~~enumerations which are value types that represent named constants.~~

24
25 2. (Canceled).

3. (Original) A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one client device.

4. (Original) A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one server device that is configured as a Web server.

5. (Currently Amended) An application program interface embodied on one or more tangible computer readable media, comprising: multiple types related to constructing user interfaces, the types comprising classes which represent managed heap allocated data that has reference assignment semantics, interfaces that define a contract that other types can implement, delegates that are object oriented function pointers, structures that represent static allocated data that has value assignment semantics and enumerations which are value types that represent named constants, wherein the application program interface is associated with a common language runtime layer that can translate Web applications written in different languages into a intermediate language supported by the common runtime layer.

1 6. (Original) An application program interface as recited in claim 5,
2 wherein the classes comprise a forms class that represents a window or a dialog
3 box that makes up an application's user interface.

4
5 7. (Original) An application program interface as recited in claim 6,
6 wherein the forms class has multiple members comprising one or more of: public
7 static properties, public static methods, public instance constructors, public
8 instance methods, public instance properties, public instance events, protected
9 instance properties, and protected instance methods.

10
11 8. (Original) An application program interface as recited in claim 5,
12 wherein the type comprising the interfaces comprises a button control interface
13 that allows a control to act like a button on a form.

14
15 9. (Original) An application program interface as recited in claim 5,
16 wherein the type comprising the interfaces comprises a container control interface
17 that provides functionality for a control to act as a parent for other controls.

18
19 10. (Original) An application program interface as recited in claim 5,
20 wherein the type comprising the interfaces comprises an editing notification
21 interface.

22
23 11. (Original) An application program interface as recited in claim 5,
24 wherein the type comprising the interfaces comprises a data object interface that
25 provides a format independent mechanism for transferring data.

1
2 12. (Original) An application program interface as recited in claim 5,
3 wherein the type comprising the interfaces comprises a feature support interface
4 that specifies a standard interface for retrieving feature information from a current
5 system.

6
7 13. (Original) An application program interface as recited in claim 5,
8 wherein the type comprising the interfaces comprises a message filter interface.

9
10 14. (Original) An application program interface as recited in claim 5,
11 wherein the type comprising the interfaces comprises a handle-exposing interface
12 to expose handles.

13
14 15. (Original) An application program interface as recited in claim 5,
15 wherein the type comprising the interfaces comprises one or more of the
16 following interfaces:

17 a button control interface that allows a control to act like a button on a
18 form;

19 a container control interface that provides functionality for a control to act
20 as a parent for other controls;

21 an editing notification interface;

22 a data object interface that provides a format independent mechanism for
23 transferring data;

24 a feature support interface that specifies a standard interface for retrieving
25 feature information from a current system;

1 a message filter interface; and

2 a handle-exposing interface to expose handles.

3
4 16. (Currently Amended) A distributed computer software architecture,
5 comprising:

6 one or more applications configured to be executed on one or more
7 computing devices, the applications handling requests submitted from remote
8 computing devices;

9 a networking platform to support the one or more applications; and

10 an application programming interface to interface the one or more
11 applications with the networking platform, the application programming interface
12 comprising various types related to constructing user interfaces, ~~wherein the~~
13 ~~various types comprise:~~

14 ~~classes which represent managed heap allocated data that has~~
15 ~~reference assignment semantics;~~

16 ~~interfaces that define a contract that other types can implement;~~

17 ~~delegates that are object oriented function pointers;~~

18 ~~structures that represent static allocated data that has value~~
19 ~~assignment semantics; and~~

20 ~~enumerations which are value types that represent named constants;~~

21 and

22 a common language runtime layer that can translate Web applications
23 written in different languages into an intermediate language supported by the
24 common runtime layer.
25

1 17. (Canceled).

2
3 18. (Currently Amended) A distributed computer software architecture
4 as recited in claim [[16]] 42, wherein the classes comprises a forms class that
5 represents a window or a dialog box that makes up an application's user interface.

6
7 19. (Original) A distributed computer software architecture as recited in
8 claim 18, wherein the forms class has multiple members comprising one or more
9 of: public static properties, public static methods, public instance constructors,
10 public instance methods, public instance properties, public instance events,
11 protected instance properties, and protected instance methods.

12
13 20. (Currently Amended) A distributed computer software architecture
14 as recited in claim [[16]] 42, wherein the type comprising the interfaces
15 comprises a button control interface that allows a control to act like a button on a
16 form.

17
18 21. (Currently Amended) A distributed computer software architecture
19 as recited in claim [[16]] 42, wherein the type comprising the interfaces
20 comprises a container control interface that provides functionality for a control to
21 act as a parent for other controls.

22
23 22. (Currently Amended) A distributed computer software architecture
24 as recited in claim [[16]] 42, wherein the type comprising the interfaces
25 comprises an editing notification interface.

1
2 23. (Currently Amended) A distributed computer software architecture
3 as recited in claim [[16]] 42, wherein the type comprising the interfaces
4 comprises a data object interface that provides a format independent mechanism
5 for transferring data.

6
7 24. (Currently Amended) A distributed computer software architecture
8 as recited in claim [[16]] 42, wherein the type comprising the interfaces
9 comprises a feature support interface that specifies a standard interface for
10 retrieving feature information from a current system.

11
12 25. (Currently Amended) A distributed computer software architecture
13 as recited in claim [[16]] 42, wherein the type comprising the interfaces
14 comprises a message filter interface.

15
16 26. (Currently Amended) A distributed computer software architecture
17 as recited in claim [[16]] 42, wherein the type comprising the interfaces
18 comprises a handle-exposing interface to expose handles.

19
20 27. (Currently Amended) A distributed computer software architecture
21 as recited in claim [[16]] 42, wherein the type comprising the interfaces
22 comprises one or more of the following interfaces:

23 a button control interface that allows a control to act like a button on a
24 form;
25

1 a container control interface that provides functionality for a control to act
2 as a parent for other controls;
3 an editing notification interface;
4 a data object interface that provides a format independent mechanism for
5 transferring data;
6 a feature support interface that specifies a standard interface for retrieving
7 feature information from a current system;
8 a message filter interface; and
9 a handle-exposing interface to expose handles.

10
11 28. (Currently Amended) A computer system including one or more
12 microprocessors and one or more software programs, the one or more software
13 programs utilizing an application program interface to request services from an
14 operating system, the application program interface including separate commands
15 to request services comprising services related to constructing user interfaces,
16 wherein the application program interface groups API functions into multiple
17 namespaces that define a collection of classes which represent managed heap
18 allocated data that has reference assignment semantics, interfaces that define a
19 contract that other types can implement, delegates that are object oriented
20 function pointers, enumerations which are value types that represent named
21 constants and structures that represent static allocated data that has value
22 assignment semantics, the application program interface being associated with a
23 common language runtime layer that can translate Web applications written in
24 different languages into an intermediate language supported by the common
25 runtime layer.

1
2 29. (Currently Amended) A method, comprising:

3 managing network and computing resources for a distributed computing
4 system; and

5 exposing a set of functions that enable developers to access the network
6 and computing resources of the distributed computing system, the set of functions
7 comprising functions to facilitate construction of user interfaces, wherein the
8 functions are grouped into multiple namespaces that define a collection of classes
9 which represent managed heap allocated data that has reference assignment
10 semantics, interfaces that define a contract that other types can implement,
11 delegates that are object oriented function pointers, enumerations which are value
12 types that represent named constants and structures that represent static allocated
13 data that has value assignment semantics; and

14 using a common language runtime layer that can translate Web
15 applications written in different languages into an intermediate language
16 supported by the common runtime layer.

17
18 30. (Original) A method as recited in claim 29, further comprising
19 receiving a request from a remote computing device, the request containing a call
20 to the set of functions.

21
22 31. (Previously Presented) A method, comprising creating a namespace
23 with functions that enable drawing and construction of user interfaces, the name
24 space defining classes which represent managed heap allocated data that has
25 reference assignment semantics, interfaces that define a contract that other types

1 can implement, delegates that are object oriented function pointers, structures that
2 represent static allocated data that has value assignment semantics, and
3 enumerations which are value types that represent named constants.

4
5 32. (Original) A method as recited in claim 31, wherein the namespace
6 defines a forms class that represents a window or a dialog box that makes up an
7 application's user interface.

8
9 33. (Original) A method as recited in claim 32, wherein the forms class
10 has multiple members comprising one or more of: public static properties, public
11 static methods, public instance constructors, public instance methods, public
12 instance properties, public instance events, protected instance properties, and
13 protected instance methods.

14
15 34. (Original) A method as recited in claim 31, wherein the namespace
16 defines an interface comprising a button control interface that allows a control to
17 act like a button on a form.

18
19 35. (Original) A method as recited in claim 31, wherein the namespace
20 defines an interface comprising a container control interface that provides
21 functionality for a control to act as a parent for other controls.

22
23 36. (Original) A method as recited in claim 31, wherein the namespace
24 defines an interface comprising an editing notification interface.

1 37. (Original) A method as recited in claim 31, wherein the namespace
2 defines an interface comprising a data object interface that provides a format
3 independent mechanism for transferring data.
4

5 38. (Original) A method as recited in claim 31, wherein the namespace
6 defines an interface comprising a feature support interface that specifies a
7 standard interface for retrieving feature information from a current system.
8

9 39. (Original) A method as recited in claim 31, wherein the namespace
10 defines an interface comprising a message filter interface.
11

12 40. (Original) A method as recited in claim 31, wherein the namespace
13 defines an interface comprising a handle-exposing interface to expose handles.
14

15 41. (New) A software architecture as recited in claim 1, wherein the
16 various types comprise classes, interfaces, delegates, structures and enumerations.
17

18 42. (New) A distributed computer software architecture as recited in
19 claim 16, wherein the various types comprise classes, interfaces, delegates,
20 structures and enumerations.
21
22
23
24
25